Hands on with puppet:

Making your cluster dance like it's on strings

Suchandra Thapa[1]

[1]Computation Institute
University of Chicago

OSG Summer Workshop, August 9-11 2011

Outline

1. Motivation
   - Why puppet?
   - Examples of places using puppet
2. The nuts and bolts
   - Basic architecture and configuration
   - Putting it all together

Outline

1. Motivation
   - Why puppet?
   - Examples of places using puppet
2. The nuts and bolts
   - Basic architecture and configuration
   - Putting it all together

Why use puppet?
Or any other configuration management system?

- Makes configuration easier.
- Allows new systems (e.g. worker nodes) to be brought online quicker.
- With appropriate additions, you can track system configuration and revert or update on a managed basis.

Why use puppet?
Or any other configuration management system?

- Makes configuration easier.
- Allows new systems (e.g. worker nodes) to be brought online quicker.
- With appropriate additions, you can track system configuration and revert or update on a managed basis.

Why use puppet?
Or any other configuration management system?

- Makes configuration easier.
- Allows new systems (e.g. worker nodes) to be brought online quicker.
- With appropriate additions, you can track system configuration and revert or update on a managed basis.

When not to use puppet

- Unique setups (e.g. web apps running on a single system)
- Systems that change rapidly (e.g. test systems)
- Software or systems that you don't feel comfortable automating

Corollary

Use puppet for services that affect large number of systems and which provide basics for these systems (e.g. ntp, ssh, user management, etc.)

When not to use puppet

- Unique setups (e.g. web apps running on a single system)
- Systems that change rapidly (e.g. test systems)
- Software or systems that you don't feel comfortable automating

Corollary

Use puppet for services that affect large number of systems and which provide basics for these systems (e.g. ntp, ssh, user management, etc.)

When not to use puppet

- Unique setups (e.g. web apps running on a single system)
- Systems that change rapidly (e.g. test systems)
- Software or systems that you don't feel comfortable automating

Corollary

Use puppet for services that affect large number of systems and which provide basics for these systems (e.g. ntp, ssh, user management, etc.)

When not to use puppet

- Unique setups (e.g. web apps running on a single system)
- Systems that change rapidly (e.g. test systems)
- Software or systems that you don't feel comfortable automating

Corollary

Use puppet for services that affect large number of systems and which provide basics for these systems (e.g. ntp, ssh, user management, etc.)

Outline

1. Motivation
   - Why puppet?
   - Examples of places using puppet
2. The nuts and bolts
   - Basic architecture and configuration
   - Putting it all together

Midwest Tier 2.

Example

Using puppet master running as a passenger app in apache
puppetd running as a cron job (due to memory leaks in client)
Puppet dashboard being added to provide monitoring of clients

BNL Tier 1

Example

Has a more complex setup:

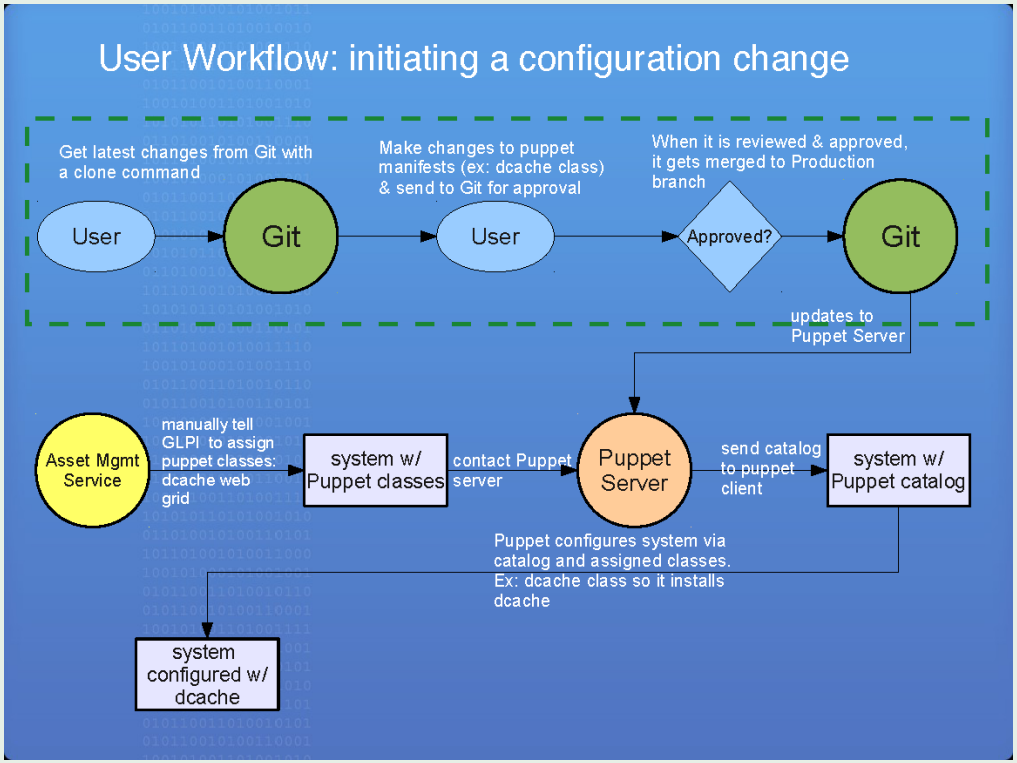

Figure: BNL Puppet Flow

Outline

1. Motivation
   - Why puppet?
   - Examples of places using puppet
2. The nuts and bolts
   - Basic architecture and configuration
   - Putting it all together

Master/Client relationship

- Puppet master (with optional dashboard) serves puppet files
- Puppet clients query server and update configuration to keep things in sync

Master/Client relationship

- Puppet master (with optional dashboard) serves puppet files
- Puppet clients query server and update configuration to keep things in sync

Basics of puppet configuration

- Node definition
- Module definition
- Templates

Node definitions

Done in nodes.pp or file included in nodes.pp

Defining a new host

```
node 'itb-test1' {
    include itb_base
    include gmond::itb::x86_64::virtual
    include user::itb::osgvo
    include user::itb::glexec
    include user::itb::condor
    include condor::submit::itb
    include yum::itb::condor::repo
}
```

Template definitions

Done in template.pp file

Defining templates

```
class itb_base::hadoop inherits itb_base {
    include user::itb::hadoop
    include yum::itb::hadoop::repo
}

class itb_base::hadoop::datanode inherits itb_base::hadoop {
    include hadoop::datanode::itb
}

class itb_base::hadoop::namenode inherits itb_base::hadoop {
    include hadoop::namenode::itb
}
```

Module definition

- Directory name same as module name
- Directories for files, manifests, templates (called `files`, `manifests`, and `templates`)
- Minimum needed is `init.pp` file in `manifests` directory

Module definition

- Directory name same as module name
- Directories for files, manifests, templates (called `files`, `manifests`, and `templates`)
- Minimum needed is `init.pp` file in `manifests` directory

Module definition

- Directory name same as module name
- Directories for files, manifests, templates (called `files`, `manifests`, and `templates`)
- Minimum needed is `init.pp` file in `manifests` directory

Condor module base class

Base condor class, installs and sets up service

Base Condor class

```
class condor::base::itb {
    package {
        'condor-7.6.2-1.x86_64' :
            ensure => present
    }

    service {
        "condor":
            enable    => true,
            ensure    => true,
            require   => [ Package["condor-7.6.2-1.x86_64"],
                           File["/etc/condor/condor_config"],
                           File["/etc/condor/condor_config.local"] ],
            subscribe => [ File["/etc/condor/condor_config"],
                           File["/etc/condor/condor_config.local"] ]
    }
}
```

Extending for worker node configuration

Sets up a condor worker node

Worker node class

```
class condor::worker::itb inherits condor::base::itb  {
    # condor configuration setup
    file {
        "/etc/condor/condor_config":
            source => "puppet:///modules/condor/condor_config",
            owner  => 'root',
            group  => 'root',
            mode   => 644
    }

    file {
        "/etc/condor/condor_config.local":
            source => "puppet:///modules/condor/condor_config_worker.local",
            owner  => 'root',
            group  => 'root',
            mode   => 644
    }
}
```

User management

Special case: In the manifest directory for the user module, users and groups defined in virtual.pp.

User and group definition

```
class user::virtual {
    @group { "glexec":
        gid => "19004",
        ensure => "present"
    }

    @user { "glexec":
        ensure  => "present",
        uid     => "19004",
        gid     => "19004",
        comment => "GLExec account",
        home    => "/home/glexec",
        shell   => "/sbin/nologin",
        require => Group['glexec']
    }
}
```

User management pt. 2

Users and groups actually get instantiated in a separate file, including the class in a node definition instantiates the user and/or group on that system

User/group instantiation

```
class user::itb::glexec inherits user::virtual {
    realize(
        Group["glexec"],
        User["glexec"]
    )
}
```

Outline

1. Motivation
   - Why puppet?
   - Examples of places using puppet
2. The nuts and bolts
   - Basic architecture and configuration
   - Putting it all together

Setting up a raw VM

1. Create modules/classes
2. Create a template for the system type
3. Create node definition for machine
4. Run puppet on systems

Summary

- If you're interested in configuration management puppet might be a solution
- Should start small and expand configuration as time goes on

- Future
  - Public repository for modules and recipes
  - Possible location for OSG collaboration?

For Further Reading I


Puppet Labs
Puppet documentation.
http://docs.puppetlabs.com/guides/introduction.html


Suchandra Thapa.
SVN Puppet repository.
http://vtb-svn.uchicago.edu/svn/puppet/


James Pryor, John Steven De Stefano Jr, Jason Alexander Smith.
Building and managing virtual machines at the Tier 1.
http://indico.cern.ch/getFile.py/access?contribId=3&resId=0&materialId=slides&confId=141745

For Further Reading II

Sarah Williams.
Experiences at MWT2: Configuration Management with Puppet.
http://indico.cern.ch/getFile.py/access?contribId=5&resId=2&materialId=slides&confId=141745